

ARMY RESEARCH LABORATORY



Real-Time News Analysis (RTNA) Scraper Assessment

by Christine E. Slocum and Ann E. M. Brodeen

ARL-TN-295

September 2007

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Aberdeen Proving Ground, MD 21005-5067

ARL-TN-295

September 2007

Real-Time News Analysis (RTNA) Scraper Assessment

Christine E. Slocum and Ann E. M. Brodeen
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE			3. DATES COVERED (From - To)	
September 2007	Final			December 2006–May 2007	
4. TITLE AND SUBTITLE				5a. CONTRACT NUMBER	
Real-Time News Analysis (RTNA) Scraper Assessment				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
Christine E. Slocum and Ann E. M. Brodeen				622783Y10	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)				8. PERFORMING ORGANIZATION REPORT NUMBER	
U.S. Army Research Laboratory ATTN: AMSRD-ARL-CI-CT Aberdeen Proving Ground, MD 21005-5067				ARL-TN-295	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT					
Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
An assessment was conducted to evaluate the performance of the Real-Time News Analysis Scraper application used to extract article body text from online news sources. The application's performance was evaluated by determining the integrity of scraped text outputted, a metric found by calculating the output's similarity to text manually selected from the same articles by a human control group. Levenshtein's edit-distance algorithm was implemented to calculate normalized similarity scores of each scraped and manually selected article text pair; normalized scores were direct indicators of integrity. The Scraper was found to perform unacceptably overall because the majority of scraped articles experienced integrity loss exceeding the established threshold. Results of the assessment were insufficiently detailed to give causal explanations for the Scraper's observed performance. Recommendations were not made for the application's improvement; however, a protocol was outlined in detail for a follow-on assessment.					
15. SUBJECT TERMS					
Real-Time News Analysis, Levenshtein, document similarity					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Ann Brodeen
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL	22	19b. TELEPHONE NUMBER (Include area code) 410-278-8947

Contents

List of Figures	iv
List of Tables	iv
1. Introduction	1
2. Performance Assessment	2
2.1 Data Sets.....	2
2.2 Performance Metrics	3
2.3 Document Preprocessing.....	3
2.4 Expectations	4
2.5 Implementation.....	4
3. Results	5
3.1 SIMILARITY TEST FAILURE Errors	6
3.2 Observed Similarity Scores.....	7
3.3 Bias.....	8
4. Recommendations	9
5. Conclusion	11
Appendix. Similarity Scores	13
Distribution List	15

List of Figures

Figure 1. Distribution of observed similarity scores into subranges.	7
---	---

List of Tables

Table 1. Breakdown of evaluated document set (797 document pairs) into observed document pairs.....	6
Table 2. Breakdown of evaluated document set (797 document pairs) into unobserved document pairs.....	6
Table A-1. Similarity score subranges.....	13
Table A-2. Unacceptable similarity scores (N = 316).	13
Table A-3. Acceptable similarity scores (N = 255).	14

1. Introduction

The Real-Time News Analysis (RTNA) initiative attempts to glean meaningful information from online news and blog articles. RTNA employs an application to automatically scrape article body text from a collection of regional and international news sources. The scraped text has been subjected to such processing as keyword filtering, translation to English, or clustering and visualization by news topic areas. These applications require the integrity of scraped articles to be fully maintained, a formidable task given the formatting markup in Web page source files.

Typically, source files are marked up in some dialect of the Standard Generalized Markup Language, such as Hyper Text Markup Language (HTML) or Extensible HTML. In these languages, elements of a page are delimited by opening and closing tags and represent nodes of a hierarchical document structure. Tags convey how a document should be displayed in a Web browser but do not describe the relationships or meanings of the formatted text, making data-aware content selection nearly impossible.

To scrape a source file, the article text must first be located in the source's document hierarchy; the text must be differentiated from nonarticle text elements. This task is made difficult by news sites that fill their article pages with menus, advertisements, pictures, captions, and embedded scripts. Such extraneous page elements provide rich visual cues outlining an article's body for a human reader, but computers have no knowledge of such implicit data descriptors and must be explicitly programmed to associate specific document structural characteristics with data characteristics.

The Scraper application, written in Java,^{*} implements the *htmlparser*[†] package to form a hierarchical representation of an article's source file in memory. In this hierarchy, each page element is a node and can be uniquely identified by its position in the document; node position is specified in terms of depth and relation to other nodes. The Scraper defines a filter[‡] for all text nested between opening and closing paragraph tags, `<p>` and `</p>`, including text further nested in children elements of these paragraphs.

The application's static approach works well if a source file's paragraph tags happen to contain article text and no extraneous page elements. However, this scenario is far from universal. Article text is frequently nested in nonparagraph nodes, in which case, the text would not be

^{*}The Scraper was written in the Java programming language and evaluated on the Java 2 Platform Standard Edition 5.0.

[†]The *htmlparser* package was obtained from <http://htmlparser.sourceforge.net/>.

[‡]Filters are classes contained in *htmlparser.filters* and define rules by which the Scraper selects article text from source files.

scraped. Such an omission is termed deletion; insertion refers to the case where nonarticle text nested in paragraph nodes is scraped, even if actual article text is also scraped. Insertions and deletions prevent the Scraper application from extracting article text of high integrity.

The assessment described in the following sections evaluated the degree to which the scraped articles experience integrity loss. Section 2 outlines the assessment’s design, implementation, and expectations. Observed results are discussed in section 3, including implications for Scraper performance and the effects of errors and bias. Section 4 recommends that a follow-up performance assessment be conducted and suggests improved design and domain sets. Section 5 provides conclusions.

2. Performance Assessment

To determine the integrity of scraped documents and with what frequency they were acceptable for further use in the RTNA application, this assessment was engineered to evaluate how well the Scraper performed when compared to a human performing comparable functionality. The data sets and performance metrics used in the assessment are described in the following subsections, as are the implementation methodologies and Scraper performance expectations.

2.1 Data Sets

One thousand news articles were compiled from over 300 unique news and blog domains and comprised the source file set. Seven hundred ninety-seven^{*} of these articles were each scraped twice (once by the RTNA Scraper and once by a human control scraper) and resulted in a scraped document and ground truth document,[†] respectively. Each scraped document was outputted by the Scraper and contained article text selected by the application; this text was scraped from a source file and saved to the scraped document directory. An arguments list was input at runtime and specified the locations of the text file’s source and the directory in which the file was saved. The scraped document set was subjected to evaluation as representation of Scraper performance.

The ground truth document set was held as the “correctly” scraped collection and, as such, was the control group; it comprised 1000 text files generated by three human scrapers. Any discrepancies between text selected by the Scraper and text selected by the human control group were attributed to Scraper error. Each source’s uniform resource locator (URL) was pasted to the first line of a new ground truth text file. A Web browser was then directed to the source’s

^{*}The Scraper was run 797 times, once for each ground truth document whose first line contained a valid, reachable URL. Section 2.5 discusses why some ground truth documents failed to meet this criterion.

[†]A scraped document and ground truth document deriving from the same source article are collectively called a document pair.

URL. From the displayed page, article body text was manually selected by a human scraper and pasted to the same text file, starting below the first line.

2.2 Performance Metrics

To evaluate how similar the Scraper's performance was to comparable human performance, an appropriate metric was necessary to evaluate how similar text automatically scraped from an article's source file was to that selected by a human viewing the same source file in a browser. The similarities of all document pairs were measured by Levenshtein's edit-distance metric, an algorithm^{*} which calculates the minimum total number of operations (insertions, deletions, or substitutions of single characters) required to transform a scraped document into its ground truth document pair.¹ The implementation of Levenshtein's algorithm used for the assessment returned a normalized, floating point similarity score between 0.0 and 1.0, inclusively, such that 0.0 indicated completely dissimilar documents and 1.0 indicated identical documents.

2.3 Document Preprocessing

The use of an edit-distance metric necessitated certain preprocessing of both document sets because of bias introduced while generating the ground truth set. New line, tab, and other whitespace string literals in the source files were placed at risk for translation and decoding differences resulting from the use of multiple text editors and browsers for compiling ground truth documents. Such differences could have caused identical characters to be incorrectly deemed dissimilar, or vice versa, by the edit-distance algorithm. Consequently, the ground truth document set was assumed to be biased because no standard dictating uniform browser and text editor usage was followed during the set's compilation.

To mitigate this likely bias, whitespace literals and other problematic characters were removed from all documents before the assessment was conducted. This removal had no significant impact on results obtained in the assessment because whitespace contributed nothing to the meaning of scraped article content, with the exception of meaning derived by word separations. Levenshtein's edit-distance algorithm determined the similarity of document pairs by comparing characters, not words, so removing whitespace between words had no negative effect on the similarity scores obtained. In fact, doing so increased the assessment's objectivity because it eliminated insignificant document differences that could have impacted similarity scores.

To remove whitespace characters, each document from the scraped and ground truth sets was encoded by UTF-8, a format chosen for its ability to represent any Unicode character and identical byte code representation of all 128 ASCII characters. ASCII, a subset of Unicode, is by

^{*}The implementation of Levenshtein's edit-distance algorithm used in this assessment was part of the *SimMetrics* package and was obtained from www.sourceforge.net/projects/simmetrics.

¹Levenshtein, V. I. *Binary Codes Capable of Correcting Deletions, Insertions, and Reversals*; Doklady Akademii Nauk SSSR, 1965, 163 (4), pp 845–848.

far the most widely used character set on the web and is accurately encoded and decoded by UTF-8. All whitespace and non-ASCII characters were removed from each document. By doing so, the impact of bias incurred during ground truth document compilation was reasonably limited.

2.4 Expectations

The authors' expectations for this assessment were modest. Based on knowledge of the Scraper's mechanics—namely, its filter for all text in paragraph nodes—highly polarized performance was expected. Each source file was predicted to be a hit or a miss; either the article text would be nested in a paragraph node or it would be nested somewhere else. The Scraper was expected to perform very well in the former case and very poorly in the latter, but there were no predictions as to how the documents would be distributed within these categories.

Among documents for which the Scraper's performance was neither very poor nor very good, insertions were expected to be more numerous than deletions, and substitutions were expected to be nearly nonexistent. Such mediocre performance would result in partially similar documents, as inserted text would likely consist of nonarticle text elements located in paragraph nodes that were scraped. If these nodes also contained article text, the scraped document would be partially similar to its corresponding ground truth document. Deletions, on the other hand, would likely occur if the article text was located in nonparagraph nodes, in which case, the scraped document would be highly dissimilar because none of the sought-after text would be selected.

Integrity loss was expected to occur because of errors in determining the location of article text within a source file. Text mislocation could feasibly cause both insertions and deletions. However, only insertions would result in mediocre performance. Substitutions would not occur from such a mislocation but would result from a character misrepresentation. The Scraper has no functionality for transforming characters, so such an error was judged unlikely to occur. Nonetheless, Levenshtein's edit-distance algorithm looks for the minimum total number of operations needed to transform a scraped document into its corresponding ground truth document. If an insertion and deletion were made at the same location in two paired documents, a substitution would be recorded by the algorithm, regardless of whether the error was actually made.

2.5 Implementation

A Python module was written to automate all evaluation tasks: preprocessing both document sets, downloading local copies of each source file, calling the Scraper while managing its input and output, calculating document similarities by invoking Levenshtein's algorithm, sorting and categorizing results, outputting results in multiple perspectives, and finally, tracking errors at each step of the assessment. In addition to these core functionalities, several scripts for

automatically graphing the results were incorporated by implementing elements of the ReportLab toolkit.*

While developing and testing the module to carry out this assessment, a number of errors in the ground truth document set came to light. Some documents were missing URLs from their first lines, and others contained correctly placed but improperly formed URLs. Both types of faults were attributed to human error. An additional group of ground truth documents contained correctly placed, valid URLs that could not be reached. The source files of these documents presumably had been removed from their previous locations or were temporarily unreachable because of network problems (e.g., web server failures or lost request packets).

Only documents which contained correctly placed, valid URLs were included in the ground truth document set. For record-keeping purposes, those documents not included in the set were assigned one of two error codes—INVALID URL or UNREACHABLE URL. The error codes SCRAPER FAILURE and SIMILARITY TEST FAILURE were devised as catch-all labels for any errors that might occur during a call to one of the embedded Java executables, Scraper and SimilarityTest.[†]

The assessment took about 3 hr to process on a higher-end Dell Pentium 4 workstation running Microsoft XP. Upon completion, several text and comma-separated value files were output to the results directory summarizing the distribution of observed[‡] similarity scores and errors of the expected document set,[§] the similarity scores of each observed document pair, the definition and numeric flag of each error code, and a list of unique domains from which the observed ground truth documents were taken.

3. Results

Similarity scores were observable for a much smaller data set than expected. Many of the original ground truth documents were unobservable due to invalid, unreachable URLs and errors in calculating pair similarities. For the remaining document pairs, similarity scores were successfully calculated and revealed clear patterns in the Scraper's performance. Table 1 summarizes the breakdown of the evaluated document set into observed and unobserved subsets. The observed document pairs were broken into five similarity score subranges, defined in the appendix (A-1). No SCRAPER FAILURE errors were recorded.

*The ReportLab Toolkit modules implemented for graphing assessment results were obtained from <http://www.reportlab.org/>.

[†]Similarity Test interfaced Levenshtein's edit-distance algorithm.

[‡]The observed document set consisted of those document pairs for which similarity scores were successfully calculated.

[§]The expected document set consisted of all ground truth documents for which the calculation of similarity scores was expected. Errors prevented the calculation of scores for many documents in this set.

Table 1. Breakdown of evaluated document set (797 document pairs) into observed document pairs.

Similarity Scores (S)	Documents
0.0 ≤ S < 0.2	102
0.2 ≤ S < 0.4	55
0.4 ≤ S < 0.6	65
0.6 ≤ S < 0.8	94
0.8 ≤ S ≤ 1.0	255

Table 2. Breakdown of evaluated document set (797 document pairs) into unobserved document pairs.

Error Code	Documents
SIMILARITY TEST FAILURE	266

A total of 203 ground truth documents caused INVALID URL and UNREACHABLE URL errors. Both errors types prevented the documents' source files from being recovered, rendering the documents useless. Each document that caused one of these errors was flagged for removal, reducing the size of the evaluated document set* to 797.

3.1 SIMILARITY TEST FAILURE Errors

SIMILARITY TEST FAILURE errors prevented the observation of an additional 226 document pairs because their similarity scores could not be calculated. Console outputs associated with each occurrence of this memory failure suggested the system's maximum recursion depth had been exceeded by the SimilarityTest process but did not indicate why. SIMILARITY TEST FAILURE errors could be explained, even in the absence of explicit error descriptions, by applying general knowledge of the dynamic programming techniques used in Levenshtein's edit-distance algorithm.

To calculate the minimum total edit-distance of two strings, s_1 and s_2 , the algorithm constructs a two-dimensional matrix with m rows and n columns where m and n are the lengths of s_1 and s_2 , respectively. Each cell, $d(i,j)$, in the matrix stores the minimum total edit-distance between the first i th characters of s_1 and the first j th characters of s_2 . The minimum total edit-distance of s_1 and s_2 , $d(m,n)$, is calculated last and stored in the bottom, right-hand cell, $d(m,n)$. The distance stored in $d(i,j)$ is defined by the recurrence relation in equation 1:

$$d(i,j) = \min \left\{ \begin{array}{l} d(i-1, j) + 1 \\ d(i, j-1) + 1 \\ d(i-1, j-1) + \text{cost} \end{array} \right\}, \text{ where } \text{cost} = \begin{cases} 0 & \text{if } s_{1,i} = s_{2,j} \\ 1 & \text{else} \end{cases}$$

$$i \in \{1, 2, \dots, m\} . \quad (1)$$

$$\text{and } j \in \{1, 2, \dots, n\}$$

*The evaluated document set consisted of all expected documents for which corresponding scraped documents were successfully obtained.

This relation defines a ternary-recursive routine² and signifies rapid exponential growth as a function of m and n. Given this time-complexity, the SIMILARITY TEST FAILURE errors were attributed to large document pair string lengths. The desktop system on which this assessment was conducted simply could not compute the edit distances of the largest document pairs, which contained many thousands of characters.

3.2 Observed Similarity Scores

The similarity scores for 571 document pairs were computed. These scores, as a whole, indicated high levels of integrity loss in most documents returned by the Scraper application and, consequently, signify unacceptable overall performance. The observed similarity scores summarized in table 1 are visually represented in figure 1.

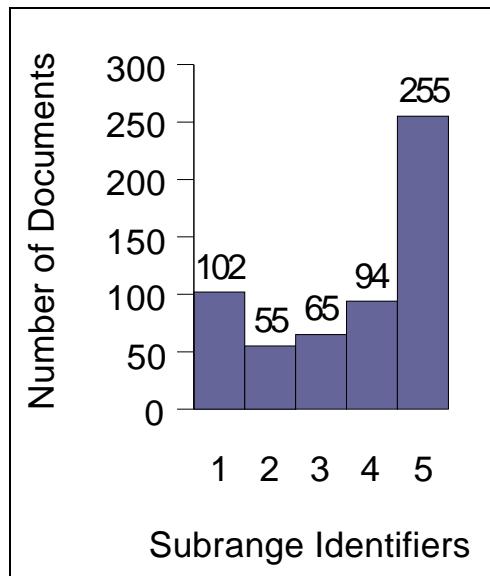


Figure 1. Distribution of observed similarity scores into subranges.

Documents with high integrity loss failed to convey concepts or keywords present in the articles and were unacceptable for use in a project for which the primary objective was information inference. A threshold was chosen specifying that a scraped document of acceptable integrity be at least 80% similar to its corresponding ground truth document. The Scraper's overall performance was acceptable only if a majority* of the scraped document set fell above the threshold.

²Allison, L. Dynamic Programming Algorithm (DPA) for Edit-Distance. <http://www.csse.monash.edu.au/~lloyd/tildeAlgDS/Dynamic/Edit/> (accessed 22 January 2007).

* A majority of the scraped document set was defined as more than half. Since the similarity scores of 571 documents were observed, the scores of at least 286 documents needed to fall above the set threshold for acceptable overall performance.

This threshold, arbitrarily chosen, may actually be too low. A scraped document omitting 20% of its source article's content could very well be unacceptable for the intended use. If most of the scraped documents fell above the threshold, the Scraper's performance could still be unacceptable, but if most fell below the threshold, the Scraper's performance would clearly be unacceptable. However, no integrity threshold was established for the content analyzed in the RTNA project, so a generous one was chosen.

By examining the observed similarity scores, it was concluded that the Scraper performed unacceptably, even though more document pairs had similarity scores in subrange 1 than any other individual subrange. Three-hundred six scraped articles experienced unacceptable integrity loss, but only 225 were acceptably scraped. More than half of the observed document pairs were found to be unacceptably dissimilar because their similarity scores fell below the 80% threshold. As this threshold was forgiving to even moderate integrity loss, the Scraper's failure to perform above it cannot be interpreted as anything but unacceptable.

The expectation of highly polarized Scraper performance was not realized because a substantial number of scraped document pairs had mediocre similarity scores that were neither highly similar nor highly dissimilar. The documents in these pairs were at least 20% and no more than 80% similar. However, the application's performance was not wholly unexpected. Similarity score subranges 1 and 5 each had larger document pair memberships than any of the other three subranges. Specifically, 357 pairs had similarity scores in subranges 1 and 5, while only 214 pairs had scores that fell in subranges 2, 3, and 4. These figures signify at least slightly polarized Scraper performance.

Expectations regarding the frequency with which specific edit operations would occur could not be verified directly because the edit-distance algorithm implemented in this assessment returned only the minimum total operations made, from which there was no way to determine the specific breakdown of insertions, deletions, and substitutions. Regardless, it was observed that many scraped documents of mediocre similarity contained more characters than their corresponding ground truth documents. This trend was consistent with the expectation that more insertions would result in mediocre similarity scores than deletions or substitutions. These scraped documents likely derived from source files whose article text was located in paragraph nodes along with extraneous text elements, all of which were scraped.

3.3 Bias

The results of this assessment were biased by inconsistent and error-prone ground truth compilation methods and by the use of a recursive edit-distance algorithm that caused a subset of the evaluated documents to be underrepresented. These sources of bias may have affected results, favorably or unfavorably, and must be considered to temper the performance implications of observed similarity scores.

The ground truth document set was compiled by three human scrapers who had different methodologies for selecting page elements to include in ground truth article text. Links, captions, advertisements, editorial notes, and time stamps were among the elements that were incorrectly included in ground truth documents. These elements often interspersed article columns and were selected nonuniformly by the human scrapers because no standard was established dictating which elements should be scraped.

The three distinct text selection methodologies used very likely biased similarity scores in this assessment, but it was not recorded which individual compiled each document. As a result, no correlations could be observed between document pair similarities and the individuals who compiled the pairs' ground truth documents. Such correlations likely existed and manifested as a higher incidence of insertions in some scraped documents and deletions in others, respectively corresponding to selection methodologies that included fewer page elements in article text and methodologies that included more.

Assessment results were further biased because the recursively implemented edit-distance algorithm failed to calculate similarity scores of long strings. The algorithm reached recursion depths that exceeded system capabilities and threw SIMILARITY TEST FAILURE errors when passed large documents. These errors prevented more than a quarter of the evaluated document set from being observed but were not evenly distributed among evaluated documents; the vast majority were caused by the largest 226 documents. Thus, similarity scores were not representative of overall Scraper performance but of performance on smaller documents only. Consequently, results were biased by the underrepresentation of a document subset on which the Scraper's performance would likely have differed.

4. Recommendations

While this assessment was primarily conducted to evaluate the Scraper's performance for RTNA, explanations as to why the application performed as it did would have enabled Scraper improvement. Insufficiently detailed results and an extremely diverse article set provided no causal explanations, so recommendations for Scraper improvements could not be made. Instead, a follow-on assessment was recommended to examine why the Scraper performed unacceptably and what modifications could be made to increase its performance.

The similarities of document pairs in each subrange reflected common structural characteristics of the pairs' source files. Based on these similarities, the Scraper's filter had similar success locating article text. Furthermore, source files sharing common characteristics had an increased likelihood of being from the same domain or similarly structured domains vs. files with few shared characteristics. Given these truths, Scraper performance could have been observed as a function of domain or in terms of the structural characteristics of source files from a domain.

Unfortunately, fewer than two observed pairs came from any single domain, so no correlations could be calculated between Scraper performance and the structural characteristics of scraped articles. Correlations could have been calculated if document pairs came from a much smaller domain set.

The edit-distance metric used was also not helpful for explaining Scraper performance. The metric only held implications for a document pair’s similarity and did not detail the minimal combination of insertions, deletions, and substitutions (the sum of which equated to the pair’s edit distance) needed to transform a scraped document into its corresponding ground truth document. Without knowledge of the specific edit operations necessary for document transformations, no inferences could be made as to whether the Scraper omitted portions of article text from imperfectly scraped documents or injected nonarticle text into them. An injection of nonarticle text would have indicated the Scraper’s filter needed refinement to more selectively scrape text from paragraph nodes. An omission of article text would have indicated the range from which the filter selected text was too restrictive and should be broadened.

Since components of the Scraper and its filter responsible for the integrity loss of scraped articles were not identified at the assessment’s conclusion, explanations could not be given for the Scraper’s poor performance, nor could recommendations for improved performance. The problematic aspects of this assessment’s design and implementation were identified, making a follow-on assessment of the Scraper’s performance the recommended course of action. Such an assessment, to enable improvement of the Scraper, must determine the causal relationships linking the application’s performance to the integrity of scraped documents.

The follow-on assessment should be modified to minimize error occurrences and to yield more useful results. Modifications should include establishing a standard for compiling the ground truth document set, restricting source articles to a set of fewer than 10 unique domains, and implementing an iterative edit-distance algorithm which would not exceed system recursion depths. Another modification should be determining a breakdown of operations made while calculating each document pair’s edit distance. An iterative algorithm implementation would necessitate certain design considerations to accommodate for the greatly increased time and space-complexities achieved during each distance calculation. The follow-on assessment would need to be conducted on a higher-performance system.

If conducted as just described, the follow-on assessment would observe similarity scores for a much higher percentage of the expected documents and include the unnormalized sums of each edit operation and the normalized similarity scores. Normalized scores would be used as they were in this assessment—to group the observed document pairs into similarity score subranges and count the number of unique domains represented by pairs in each subrange. Unnormalized operation sums would then be used to correlate the structural characteristics of observed document pair source files and the subranges into which the pairs were grouped.

5. Conclusion

Given the project's objectives, the Scraper application's performance was deemed unacceptable for use by RTNA because scraped online news articles observed in the assessment experienced unacceptable integrity loss. It was not determined why the Scraper performed as it did because the implemented edit-distance algorithm did not detail the errors responsible for integrity loss and was unable to evaluate the largest scraped articles. Also, the set of unique domains from which the observed articles derived was too large for any correlations to be observed between the structural characteristics of source files and the integrity of article text scraped from them.

Improvements to the application could not be recommended because causes of the Scraper's unacceptable performance were not identifiable by the analysis tool. A follow-on assessment is needed to further investigate why the Scraper incorrectly selected the article text of so many observed source files.

INTENTIONALLY LEFT BLANK.

Appendix. Similarity Scores

A.1 Similarity Score Subranges

Document pair similarity scores are normalized floating point numbers between 0.0 and 1.0. A score of 0.0 indicates a completely dissimilar document pair, while a score of 1.0 indicates two identical documents. For the purpose of analyzing the Scraper's performance, similarity scores, S, of all observed document pairs were grouped into five subranges, each 20% of the inclusive range 0.0 to 1.0 (see table A-1).

Table A-1. Similarity score subranges.

Subrange Identifiers	Subranges
1	$0.0 \leq S < 0.2$
2	$0.2 \leq S < 0.4$
3	$0.4 \leq S < 0.6$
4	$0.6 \leq S < 0.8$
5	$0.8 \leq S \leq 1.0$

A.2 Similarity Score of Observed Document Pairs

Similarity scores, S, of all observed document pairs are grouped into unacceptable scores (table A-2) such that $0.0 \leq S < 0.8$ and acceptable scores (table A-3) such that $0.8 \leq S \leq 1.0$. N is the number of document pairs in each group.

Table A-2. Unacceptable similarity scores (N = 316).

0.01	0.03	0.07	0.09	0.14	0.17	0.19	0.21	0.25	0.33	0.41	0.45	0.51	0.56	0.62	0.67	0.70	0.73	0.77	0.78
0.01	0.04	0.07	0.10	0.14	0.17	0.19	0.21	0.26	0.34	0.41	0.45	0.52	0.57	0.62	0.67	0.70	0.73	0.77	0.78
0.01	0.04	0.07	0.10	0.15	0.17	0.19	0.21	0.27	0.34	0.42	0.45	0.52	0.57	0.62	0.67	0.70	0.74	0.77	0.79
0.01	0.04	0.08	0.10	0.15	0.17	0.19	0.22	0.27	0.34	0.42	0.45	0.52	0.58	0.63	0.67	0.71	0.74	0.77	0.79
0.01	0.04	0.08	0.10	0.15	0.18	0.20	0.22	0.28	0.34	0.42	0.46	0.52	0.58	0.63	0.67	0.71	0.74	0.77	0.79
0.01	0.04	0.08	0.10	0.15	0.18	0.20	0.22	0.29	0.35	0.42	0.46	0.53	0.58	0.64	0.68	0.71	0.74	0.77	0.79
0.01	0.04	0.08	0.11	0.16	0.18	0.20	0.22	0.29	0.35	0.43	0.47	0.53	0.59	0.64	0.68	0.71	0.74	0.77	0.79
0.01	0.05	0.08	0.11	0.16	0.18	0.21	0.22	0.30	0.36	0.43	0.48	0.53	0.59	0.64	0.68	0.71	0.74	0.78	0.80
0.01	0.05	0.08	0.11	0.16	0.18	0.21	0.22	0.30	0.36	0.43	0.48	0.53	0.59	0.64	0.69	0.71	0.74	0.78	0.80
0.01	0.05	0.08	0.11	0.16	0.18	0.20	0.22	0.29	0.35	0.43	0.47	0.53	0.59	0.64	0.68	0.71	0.74	0.78	0.80
0.01	0.05	0.08	0.11	0.16	0.18	0.21	0.23	0.30	0.37	0.43	0.49	0.54	0.59	0.65	0.69	0.71	0.74	0.78	0.80
0.01	0.05	0.08	0.12	0.16	0.18	0.21	0.23	0.30	0.38	0.44	0.50	0.54	0.59	0.65	0.69	0.72	0.74	0.78	0.80
0.01	0.06	0.09	0.13	0.16	0.18	0.21	0.23	0.32	0.39	0.44	0.50	0.54	0.59	0.65	0.69	0.72	0.75	0.78	0.80
0.02	0.06	0.09	0.13	0.16	0.19	0.21	0.24	0.32	0.40	0.44	0.50	0.55	0.60	0.65	0.69	0.72	0.76	0.78	—
0.02	0.06	0.09	0.13	0.16	0.19	0.21	0.24	0.32	0.40	0.44	0.50	0.55	0.60	0.66	0.69	0.72	0.76	0.78	—
0.02	0.06	0.09	0.14	0.17	0.19	0.21	0.24	0.32	0.40	0.44	0.51	0.55	0.61	0.66	0.69	0.72	0.76	0.78	—
0.03	0.07	0.09	0.14	0.17	0.19	0.21	0.25	0.32	0.40	0.45	0.51	0.56	0.61	0.66	0.69	0.72	0.76	0.78	—

Table A-3. Acceptable similarity scores (N = 255).

0.80	0.82	0.83	0.84	0.85	0.87	0.88	0.88	0.89	0.91	0.91	0.92	0.93	0.93	0.94	0.95	0.96	0.97	0.98	1.00
0.80	0.82	0.83	0.84	0.85	0.87	0.88	0.88	0.89	0.91	0.91	0.92	0.93	0.93	0.94	0.95	0.96	0.97	0.98	1.00
0.81	0.82	0.83	0.85	0.85	0.87	0.88	0.88	0.89	0.91	0.91	0.92	0.93	0.93	0.94	0.95	0.96	0.97	0.98	1.00
0.81	0.82	0.83	0.85	0.85	0.87	0.88	0.88	0.90	0.91	0.91	0.92	0.93	0.93	0.94	0.95	0.96	0.97	0.98	1.00
0.81	0.82	0.83	0.85	0.85	0.87	0.88	0.88	0.90	0.91	0.91	0.92	0.93	0.93	0.94	0.95	0.97	0.97	0.98	1.00
0.81	0.82	0.83	0.85	0.85	0.87	0.88	0.89	0.90	0.91	0.91	0.92	0.93	0.93	0.95	0.95	0.97	0.97	0.98	1.00
0.81	0.82	0.83	0.85	0.86	0.87	0.88	0.89	0.90	0.91	0.91	0.92	0.93	0.93	0.95	0.96	0.97	0.97	0.98	1.00
0.81	0.82	0.84	0.85	0.86	0.87	0.88	0.89	0.90	0.91	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.97	0.98	1.00
0.81	0.82	0.84	0.85	0.86	0.87	0.88	0.89	0.90	0.91	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.98	—
0.81	0.82	0.84	0.85	0.86	0.87	0.88	0.89	0.90	0.91	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99	—
0.81	0.83	0.84	0.85	0.86	0.87	0.88	0.89	0.90	0.91	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99	—
0.82	0.83	0.84	0.85	0.86	0.88	0.88	0.89	0.90	0.91	0.91	0.93	0.93	0.94	0.95	0.96	0.97	0.98	0.99	—
0.82	0.83	0.84	0.85	0.86	0.88	0.88	0.89	0.91	0.91	0.92	0.93	0.93	0.94	0.95	0.96	0.97	0.98	0.99	—

NO. OF
COPIES ORGANIZATION

1 DEFENSE TECHNICAL
(PDF INFORMATION CTR
ONLY) DTIC OCA
8725 JOHN J KINGMAN RD
STE 0944
FORT BELVOIR VA 22060-6218

1 US ARMY RSRCH DEV &
ENGRG CMD
SYSTEMS OF SYSTEMS
INTEGRATION
AMSRD SS T
6000 6TH ST STE 100
FORT BELVOIR VA 22060-5608

1 DIRECTOR
US ARMY RESEARCH LAB
IMNE ALC IMS
2800 POWDER MILL RD
ADELPHI MD 20783-1197

3 DIRECTOR
US ARMY RESEARCH LAB
AMSRD ARL CI OK TL
2800 POWDER MILL RD
ADELPHI MD 20783-1197

ABERDEEN PROVING GROUND

1 DIR USARL
AMSRD ARL CI OK TP (BLDG 4600)

NO. OF
COPIES ORGANIZATION

4 DIR USARL
AMSRD ARL CI CB
C VOSS
R HOBBS
C TATE
J MICHER
ADELPHI MD 20783-1197

ABERDEEN PROVING GROUND

24 DIR USARL
AMSRD ARL CI CT
A BRODEEN (10 CPS)
J BRAND
J DUMER
J FORESTER
J O'MAY
C SLOCUM (10 CPS)